

Universiteit Leiden

Master Computer Science

Do Generative AI Agents Need a Manager? Extending Organizational Coordination Theory to Artificial Teams

Name:	Burak Agachan
Student ID:	4186095
Date:	26/06/2026
Specialisation:	Data Science
1st supervisor:	Zohrehvand, A.H. (Miros)
2nd supervisor:	Duijn, M.J. van (Max)

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Contents

Acknowledgements 4

Abstract 5

1 Introduction 6

2 Theoretical Background 7

2.1 Organizational Coordination Theory . . . 7
2.2 Flat and Hierarchical Coordination in Multi-Agent Systems . . . 8
2.3 Hypotheses . . . 9

3 Method 10

3.1 Research Setting . . . 10
3.2 Procedure . . . 12
3.3 Measures . . . 13
3.4 Analytical Strategy . . . 14

4 Results 14

4.1 Test of H1a vs H1b: Classical Exploitation Prediction versus LLM-Specific Degradation . . . 14
4.2 H2: Flat Coordination Produces Higher Utility . . . 16
4.3 Inter-Rater Reliability . . . 16
4.4 Efficiency Trade-offs . . . 17
4.5 Is the Harm from Hierarchy Itself, or from its Revision Loops? . . . 18
4.5.1 Dose–Response within the Hierarchical Arm . . . 18
4.5.2 Topology Effect Isolated at Zero Loops . . . 19

5 Discussion 21

5.1 Summary of Findings . . . 21
5.2 Contributions to Multi-Agent LLM Systems Research . . . 22
5.3 Contributions to Organizational Coordination Theory . . . 23
5.4 Limitations and Boundary Conditions . . . 24
5.5 Practical Implications . . . 25
5.6 Conclusion . . . 25

6 Data Sources and Ethics 26

6.1 Amazon review and product-specification data . . . 26
6.2 AI-use disclosure . . . 26
6.3 Data and code availability . . . 27

A Robustness Checks: Wilcoxon and Holm-Bonferroni Correction **28**

B Per-Judge Directional Concordance **29**

C Architectural Main-Effect Mixed-Effects Regression **29**

D Paired-Difference Inter-Judge Correlations **30**

E Construct Validity: Inter-Dimension Correlations and Rubric Anchors **31**

F Hierarchical Manager Prompt (Verbatim) **34**

Acknowledgements

I would like to express my sincere gratitude to my first supervisor, Dr. A.H. Zohrehvand, for his invaluable guidance, patience, and intellectual rigour throughout every stage of this research. His feedback shaped not only the methodology and framing of this thesis but also the discipline with which I learned to make and defend empirical claims. I am especially thankful for the depth and care he brought to every revision round; his presence as my supervisor genuinely made the difference between a finished thesis and one I am proud of.

I also sincerely thank my second supervisor, Dr. M.J. van Duijn, for his thoughtful feedback at key moments of this work. His questions and suggestions sharpened the presentation and helped me see the thesis from the perspective of a reader meeting it for the first time. His involvement is genuinely appreciated.

I am also deeply grateful to my wife, Hazal, whose patience and steady support throughout the long stretches of this work were a constant source of strength. To my son, Aren, who arrived during this journey and gave it a meaning beyond the thesis itself. I would also like to thank my family and friends for their encouragement, their honest interest in the work, and the perspective they brought when I most needed it. Their belief and continued presence throughout this process are deeply appreciated.

Abstract

This study explores if theories of flat and hierarchical influence structures transfer from human organizations to LLM agent teams. The study implements two architectures sharing the same five agents, prompts, tools, and data, differing only in whether the Manager can reject worker output and force a revision loop. Using a sample of e-commerce products, the study tests how teams of agents generate and evaluate information into coherent reports. Three findings emerge. First, flat coordination produces significantly higher strategic Utility, preserving the exploratory reasoning the classical theory predicts. Second, hierarchical coordination produces wordier reports without improving structural clarity, so the Writing Clarity advantage for Flat reflects verbosity introduced by revision loops rather than the decisive convergence the classical theory expects. Third, specification-extraction accuracy does not differ between architectures and is instead governed by writer-model identity and evidence availability. A within-hierarchical analysis shows each revision loop is associated with a drop of roughly 0.14 Writing Clarity points (correlational, since the Manager chooses the loop count), consistent with iterative revision as the harm mechanism, while hierarchical coordination also consumes 51.5% more tokens for no quality gain. Together, the study finds that the organizational theory of human teams transfers asymmetrically to artificial teams: exploration survives, but iterative revision under authoritative prompting results in poor outcomes rather than deliberation.

1 Introduction

Multi-agent LLM systems are increasingly deployed for short-context knowledge synthesis, and the dominant architectural pattern in production frameworks is hierarchical: a Manager agent sits above worker agents and reviews their output, on the practitioner intuition that managerial oversight improves quality. This pattern aligns with classical organizational coordination theory, in which hierarchical influence structures are predicted to drive faster convergence on high-quality outputs [3]. The same pattern, however, sits in direct tension with a second body of evidence from the LLM literature: language models comply sycophantically with authoritative prompts [21], a phenomenon documented in the broader RLHF-training literature [22] and are susceptible to Degeneration-of-Thought (DoT) under iterative critique [26], both of which predict that hierarchical revision in an LLM team should produce worse output, not better. The three views (practitioner default, classical theory, and LLM-specific concern) generate incompatible predictions, and a controlled head-to-head adjudication between them is largely absent. This paper provides one. Connecting multi-agent LLM systems research with organizational coordination theory, we ask whether predictions originally derived for human teams transfer to teams of AI agents, and we let the data choose between the competing predictions.

On the systems side, frameworks such as MetaGPT [6], AutoGen [4], and AgentOrchestra [5] have established that specialised agent teams can improve output quality, and recent surveys treat hierarchical orchestration with managerial oversight as the default architecture for complex knowledge work [32]. On the organizational side, Malone and Crowston [1] define coordination as managing dependencies between activities, and Kocak, Levinthal, and Puranam [3] formalise this through a framework in which the influence structure (whether agents are arranged flat or hierarchical) is the decisive factor in team performance, with hierarchies driving exploitation and flat structures preserving exploration.

A controlled head-to-head test must hold everything else constant (the agents, the prompts, the tools, the data, and the task) and vary only the structural variable Kocak et al.'s framework actually targets: whether a Manager has the authority to reject worker output and force revision. The need for this test is sharpened by the fact that LLM agents differ from human team members in ways directly relevant to the theory's assumptions. They generate tokens sequentially and cannot replicate the parallel broadcast that defines Kocak et al.'s flat topology. They are documented to lose divergent reasoning and retreat to hedged, repetitive text when repeatedly critiqued by an authoritative source (Degeneration-of-Thought, DoT) [26], and to comply sycophantically with authoritative prompts [21, 22]. Together, these properties suggest that the very feedback loops that drive hierarchical coordination in human teams may degrade rather than improve LLM output quality. Whether that is what actually happens, and whether the failure mode is

structural (the topology itself) or mechanistic (the iterative revision the topology enables), are the questions the present study is built to answer.

This study addresses the gap by implementing and empirically comparing two LLM-based multi-agent architectures for automated Business Intelligence report generation: a flat pipeline (control) and a hierarchical system with Socratic managerial oversight (treatment). Both architectures share the same five agents, prompts, tools, and data access, differing only in the Manager's *loop-back authority* (the power to reject a worker's output and force the previous worker to revise; absent in the flat control, present in the hierarchical treatment with a cap of two loops). From an initial pool of 52 collected products, 43 met our $\geq$5-review threshold and entered the paired analysis, evaluated with a 5-model LLM judge panel and deterministic accuracy checking. We test two competing predictions for Writing Clarity, drawn from the two literatures: a classical prediction grounded in Kocak et al.'s [3] framework that hierarchical coordination produces higher Writing Clarity through exploitation-driven convergence (H1a), versus an LLM-specific prediction that it instead produces lower Writing Clarity through Degeneration-of-Thought and sycophantic compliance under iterative revision (H1b). For Utility we hypothesise that flat coordination preserves higher scores by protecting exploratory reasoning (H2). The paper contributes (i) for multi-agent LLM systems research, by isolating the iterative revision loop as the dominant mechanism driving quality differences and showing that a flat sequential pipeline is a strong default; (ii) for organizational coordination theory, by identifying two boundary conditions that explain why Kocak et al.'s framework transfers asymmetrically to LLM teams; and (iii) methodologically, through a hybrid evaluation framework combining subjective LLM judge scoring with a deterministic specification-extraction script that verifies recovery of ground-truth product specs.

2 Theoretical Background

2.1 Organizational Coordination Theory

Our system architecture is grounded in three frameworks regarding task interdependence and multi-agent information flow. The Transformer architecture [7] enabled large-scale language models exhibiting emergent capabilities such as in-context learning [8] and multi-step reasoning [9], transforming LLMs into cognitive engines that can power autonomous agents. Orchestrating such agents, however, requires coordination theory.

Malone and Crowston [1] define coordination as managing dependencies between activities; for multi-agent systems, the critical question is whether the flow of information between agents can satisfy the dependencies of the overall goal. Simon [2] adds the structural logic of near-decomposability: complex systems arrange themselves into hierarchies where intra-subsystem interactions dominate, supporting our decomposition of BI report

generation into research, analysis, writing, and critique.

Kocak, Levinthal, and Puranam [3] provide a dynamic theory of how those structures adapt and learn. Their framework addresses the dual challenge: organizations need to search for better solutions while simultaneously coordinating to align actions across the team. They model this through influence structures, representing the network topologies through which agents share and update beliefs, identifying three types: the crowd (isolated agents), the flat team (dense symmetric influence), and the hierarchical team (asymmetric influence where a leader directs the group).

Kocak et al.'s key insight concerns how these structures affect the resolution and similarity of agent beliefs. Flat teams achieve high similarity but low resolution: agents converge quickly on vague, indecisive preferences. Hierarchical teams achieve both, anchored by the leader's stable belief, but risk entrapment on non-peaks, converging so fast on a suboptimal answer that they stop searching. Whether these dynamics transfer from human organizations to LLM agent teams is the central empirical question of this study.

A critical structural difference motivates our investigation: Kocak et al.'s flat team is modeled as a fully connected, simultaneous network, while LLM agents produce outputs sequentially within a context window and process accumulated shared state rather than maintaining persistent beliefs that update through parallel peer interaction. This asymmetry motivates testing which aspects of the theory survive translation to sequential, token-based systems.

What we preserve from Kocak et al.'s framework is their core independent variable: influence asymmetry, defined as whether any single agent holds the authority to reject and force revision of another agent's output. Because LLMs cannot replicate the simultaneous fully-connected flat topology, our Flat architecture does not claim to instantiate symmetric network influence; it operationalizes the absence of coercive top-down loop-back authority. Our Hierarchical architecture implements asymmetric influence: a Manager issues directive critiques and holds mandatory loop-back authority.

2.2 Flat and Hierarchical Coordination in Multi-Agent Systems

In flat coordination, all agents sit at the same level with no designated supervisor; coordination emerges from message exchange and shared state. Tran et al. [32] systematize these mechanisms across four paradigms (memory, report, relay, debate) and four topologies (bus, star, ring, tree), confirming that topology is a first-order design decision. Through Kocak et al.'s lens, flat teams achieve rapid similarity but at low resolution: without an asymmetric influencer, peer agents tend to drift toward vague, averaged positions.

The hierarchical architecture rests on four theoretical pillars: strategic discrimination (a Manager forces workers to develop clearer positions through specific constraints), reasoning depth (iterative directive revision, inspired by Socratic Iterative Reasoning [13],

in which the Manager flags logical gaps and forces the responsible worker to redo the step), hallucination mitigation (a Quality Gate via the Critic–Manager loop, drawing on CRITIC [14]), and cognitive load management (targeted instructions mitigate the long-context limitation [12]). However, as Kocak et al. warn, hierarchies carry the inherent risk of entrapment on non-peaks.

2.3 Hypotheses

Our self-correction separates generation and critique into distinct agents with external verification. Inspired by Reflexion [15], the Critic uses a verification tool to cross-reference assertions against a vector database before the Manager routes, preventing the echo chamber where a single model confirms its own hallucinations. A key coordination challenge is error propagation: in flat systems, an error introduced by one agent tends to propagate downstream unchecked, while hierarchical systems use the Manager as a checkpoint at the cost of additional tokens, money, and latency.

Designing effective multi-agent AI systems requires balancing discovery with synthesis, mirroring the exploration–exploitation trade-off [25]. Kocak et al. [3] demonstrate that flat teams stimulate broad exploration but struggle to converge, while hierarchical teams drive rapid exploitation at the risk of entrapment. In LLM systems, this convergence manifests through Degeneration-of-Thought (DoT) [26]: under iterative feedback, agents lose the capacity to generate divergent thoughts and retreat to safe, hedged text, since RLHF-trained models exhibit structural sycophancy when corrected by an authoritative prompt [21, 22].

The hypotheses for Writing Clarity reflect a tension between these two literatures. Drawing on organizational coordination theory, the classical expectation is that hierarchical structures excel at exploitation by driving rapid convergence on high-resolution beliefs [3]. Under this view, granting the Manager loop-back authority should enhance text quality through strategic discrimination (specific constraints that force subordinate agents to develop clearer positions) and iterative revision (a progressive polishing of the report over multiple passes [13]), with the Critic–Manager loop acting as a quality gate [14]. The LLM-specific properties discussed above, however, predict the opposite outcome: iterative feedback under authoritative prompting induces Degeneration-of-Thought [26] and sycophantic compliance [21, 22], so coercive revision is more likely to produce hedged, verbose text than sharpened reasoning. We formalise this empirical tension as two competing predictions and let the data adjudicate between them:

H1a (Classical): Compared to flat coordination, hierarchical coordination will produce reports with significantly higher Writing Clarity.

H1b (LLM-Specific): Compared to flat coordination, hierarchical coordination will produce reports with significantly lower Writing Clarity, driven by Degeneration-

of-Thought and sycophantic compliance under iterative revision.

Turning to Utility, flat coordination structures inherently preserve exploration by allowing agents to maintain divergent ideas without the pressure of forced convergence [3]. Without the threat of top-down rejection from a Manager, agents within an unmanaged pipeline can engage in broader, more exploratory analysis that yields deeper strategic insights. This non-coercive environment directly mitigates the risk of Degeneration-of-Thought, preventing the system from prematurely converging on suboptimal narratives under iterative review [26], and avoids triggering the sycophantic compliance prevalent in RLHF-trained models [22], so agents preserve their original analytical rigour [21] instead of retreating into safe, hedged text. We therefore expect this structure to protect the strategic value of the reports:

H2: Compared to hierarchical coordination, flat coordination will produce reports with significantly higher Utility.

3 Method

3.1 Research Setting

Business Intelligence reporting within e-commerce serves as our experimental testbed. Modern marketplaces generate enormous volumes of unstructured customer feedback, making automated synthesis operationally necessary [19]. This task captures Kocak et al.'s [3] dual challenge of search (finding relevant facts) and coordination (aligning them into a coherent narrative). The dual nature of the data (verifiable specs plus subjective reviews) lets us evaluate both specification-extraction Accuracy and strategic Utility, the latter justified by Doshi et al.'s [20] finding that generative AI can reliably evaluate strategic decisions.

We collected data from Amazon US marketplace, targeting gaming laptops with the RTX4060 GPU. Focusing on a single GPU category controls for hardware heterogeneity. After cleaning (removing duplicates and filtering short reviews), the candidate pool contained 52 products with 540 reviews. We applied a $\geq$5-review threshold; nine products fell below it and were excluded, leaving 43 paired products. Reviews are stratified by sentiment via star ratings (4+ = positive, $\leq$3 = negative) to prevent positive-skew bias.

We built the system using LangGraph [23], modelling multi-agent workflows as state machines. Each worker agent is instantiated using the ReAct paradigm [10], interleaving reasoning with tool actions. To ensure findings are not biased by a single model family, a Random Allocation Pool of five state-of-the-art models is used: GPT-5.4, Gemini-3.1-Pro, Qwen-3.5-122B, GLM-5, and Mistral-Large-3. Worker roles draw from the full pool; the Manager role draws from a four-model subset (Qwen excluded, as its Thinking-Mode output stream is incompatible with the strict structured-output schema used for

routing). The same Manager-eligible pool is used in both conditions per product, so this asymmetry does not interact with the treatment. Temperature is set to 0; we additionally pin the API endpoint version per provider and record the exact model identifier returned. Data is stored in ChromaDB [28] with Sentence-BERT embeddings [27], implementing a Retrieval-Augmented Generation (RAG) design [11].

Table 1: System Architecture Components and Technologies

Component	**Technology**	**Key Reference**
Orchestration	LangGraph + LangChain	ReAct paradigm [10]
LLM (Report Gen.)	5 SOTA Models (Random Pool)	Multi-model robustness
LLM (Evaluation)	5-Judge Panel	[24]
Vector Database	ChromaDB	[28]
Embeddings	all-MiniLM-L6-v2	Sentence-BERT [27]
Data Source	Amazon Product Data	52 collected, 43 retained, 540 reviews

Three tools are wired to the experimental agents (the Researcher uses `get_product_specs` for metadata retrieval and `search_reviews` for semantic search with sentiment filtering; the Critic uses `verify_claim` for hallucination detection via similarity scoring). A fourth tool, `get_product_list`, exists in the codebase for interactive Streamlit use but is not bound to any agent in the experimental pipeline, since the target ASIN is supplied programmatically. The verification tool converts L2 distance to a similarity score:

$$\text{similarity}(q, d) = \frac{1}{1 + \|q - d\|_2} \quad \in [0, 1] \tag{1}$$

with thresholds: ≥ 0.7 = VERIFIED, 0.5–0.7 = PARTIAL, < 0.5 = UNVERIFIED. The 0.5/0.7 cut-points were calibrated heuristically on a pilot inspection of representative claim–evidence pairs drawn from outside the analytic sample; the labels are reported to the Critic and Manager but are not used as numeric inputs to any downstream metric, limiting the impact of any specific threshold choice.

The core team consists of four specialized agents, identical across architectures: Researcher (data collection, no interpretation), Analyst (pattern identification and root-cause analysis, no tools), Writer (professional report with required Technical Specifications section), and Critic (tool-interactive critiquing via `verify_claim`). All prompts are identical across architectures; any performance difference is strictly due to coordination structure.

Flat Architecture (Control): An unmanaged linear pipeline built upon a Shared Blackboard. Because LLMs cannot replicate Kocak et al.'s simultaneous, fully-connected

flat topology, the Flat architecture does not attempt to instantiate symmetric network influence; instead, it operationalizes the single structural variable we manipulate: the absence of a coercive supervisor with loop-back authority. A Peer Manager node acts as a non-coercive routing dispatcher, enforcing strict linear workflow: Researcher → Analyst → Writer → Critic → FINISH. The `loop_count` is hardcoded to zero, so no agent possesses authority to reject work or force revision.

Hierarchical Architecture (Treatment): The Manager is granted loop-back authority; on identifying quality issues, it writes binding directives injected into the worker's context. The Manager's prompt operationalises asymmetric influence through a structured workflow: after each worker step, the Manager reviews the Blackboard, applies an explicit decision guide (e.g., for the Researcher output, "Does the data include product specs AND customer reviews with quotes?"), and either proceeds to the next worker or issues a binding loop-back directive. A Reflexion safety net has the Critic run `verify_claim` against the database, and the Manager routes to END only if the APPROVED flag is present. Maximum loop count is set to 2. The full Manager prompt is reproduced in Appendix F.

$$\text{status}(c) = \begin{cases} \text{UNVERIFIED} & \text{if similarity}(c) < 0.5 \\ \text{PARTIAL} & \text{if } 0.5 \leq \text{similarity}(c) < 0.7 \\ \text{VERIFIED} & \text{if similarity}(c) \geq 0.7 \end{cases} \tag{2}$$

Table 2: Comparison of Coordination Architectures

Feature	**Flat (Control)**	**Hierarchical (Treatment)**
Agents	4 + Router	4 + Manager
Coordination	Sequential routing	Directive management with loop-back
Feedback Loops	None (linear pipeline)	Manager can loop back (max 2)
Verification	Critic only	Critic + Manager loop-back
State Passing	Shared Blackboard (no loop-back)	Shared Blackboard + targeted instructions
Max Steps	9	15

3.2 Procedure

Both architectures generate one BI report per product under identical conditions. For each product, a random model assignment is applied identically to both architectures, ensuring a strictly controlled paired comparison. An efficiency tracker records tokens,

API cost, and end-to-end latency, after which each report enters the evaluation engine for hybrid scoring.

3.3 Measures

The LLM-as-a-Judge paradigm uses state-of-the-art models to assess text quality. The G-EVAL framework [16] showed that LLMs with detailed criteria produce scores aligning well with human judgment, though LLM judges carry position, verbosity, and self-enhancement biases [17]. We mitigate these via a multi-model judge panel for subjective metrics and a deterministic specification-extraction script for verifiable factual content. For specification-extraction accuracy, we adapt the FActScore framework [18]: rather than checking arbitrary atomic facts (which would require an LLM verifier and reintroduce recursive hallucination risk), we restrict verification to technical specifications that can be matched against ground-truth product metadata via regex patterns. This deliberately narrows the scope of what we measure: we measure whether reports correctly recover verifiable specs, not the factuality of analytical claims. Following Choudhury, Vanneste, and Zohrehvand's [24] qualitative-first methodology, our judges produce detailed qualitative analysis before assigning scores.

We use a 5-Judge Panel (GPT-5.4, Gemini-3.1-Pro, Qwen-3.5-122B, GLM-5, Mistral-Large-3) following the Wade Test methodology [24]. Each judge produces Chain-of-Thought analysis before assigning scores from 1 to 5. Writing Clarity (proxy for exploitation [25]) is measured through Structure, Coherence, Conciseness; Utility (proxy for exploration) through Actionability, Root Cause Analysis, Strategic Depth. The mapping between these rubric dimensions and the exploitation/exploration constructs is an interpretive assumption of this study, not a validated correspondence: Structure is the per-Writing-Clarity dimension most directly aligned with Kocak et al.'s notion of belief convergence and decisiveness, and Strategic Depth is the per-Utility dimension most directly aligned with exploratory reasoning, but we did not formally validate the mapping through cross-instrument comparisons. We return to the implications of this assumption in §5.3. Specification-extraction accuracy (abbreviated as Accuracy in tables and equations) uses a deterministic script inspired by FActScore [18]: regex patterns extract technical specifications and compare against ground-truth product metadata, with the recovery ratio scaled to a 1–5 axis. This metric tests whether reports correctly recover verifiable specs; it does not assess the factuality of analytical claims.

The overall score is the arithmetic mean of the three pillars:

$$\text{Final Score} = \frac{\text{Writing Clarity} + \text{Utility} + \text{Accuracy}}{3} \tag{3}$$

3.4 Analytical Strategy

Our primary tests are paired-sample t-tests on Writing Clarity (which adjudicates between H1a and H1b) and Utility (H2), with the paired-sample variant of Cohen's d (mean of paired differences over the SD of differences) as the effect-size index. To trace the mechanism behind any topology effect, we additionally fit a within-hierarchical dose-response regression of quality scores on LoopCount, controlling for writer model, number of reviews, and mean rating; because LoopCount is mechanically zero in Flat by construction, it cannot be entered alongside Architecture in a single regression. We complement this with a paired comparison restricted to hierarchical runs that finished in zero loops, where any remaining difference is attributable to topology rather than to revision. Inter-rater reliability is reported via Krippendorff's alpha [29]. As robustness checks, all primary tests are repeated with Wilcoxon signed-rank alternatives and with Holm-Bonferroni step-down correction [30], and an architectural main-effect mixed-effects regression (Architecture + controls, product as random intercept) is fit on the full $N = 86$ sample; full results are in the Appendix.

With $N = 43$ paired observations and a two-sided paired t-test at $\alpha = 0.05$, our design achieves approximately 80% power to detect a medium effect size ($d = 0.44$) and approximately 65% power for a small-to-medium effect ($d = 0.35$). The paired design further increases effective power by eliminating between-product variance, as each product serves as its own control. We do not report observed-power calculations for the realised effect sizes, since post-hoc power computed at observed effects is statistically circular [31] and adds no information beyond the p-value. The per-dimension follow-up tests within Writing Clarity and Utility (six dimensions) are individually underpowered for the small effects they probe and are reported as exploratory; the primary hypotheses operate at the composite level, where the a priori power calculation above applies.

4 Results

We conducted 86 experimental runs (43 Flat + 43 Hierarchical). Every run completed successfully. The total experiment cost $47.71 in API fees and took approximately 5.9 hours.

At the product level, the Flat architecture won on Final Score in 28 out of 43 products (65%), while the Hierarchical architecture won in 14 (33%), with one tie.

4.1 Test of H1a vs H1b: Classical Exploitation Prediction versus LLM-Specific Degradation

The classical prediction (H1a) is rejected in favour of the LLM-specific prediction (H1b): Flat scores higher than Hierarchical on Writing Clarity (Cohen's $d = 0.342$, paired-t

Table 3: Overall Score Comparison (N = 43 paired products)

Metric	**Flat (M ± SD)**	**Hierarchical (M ± SD)**	**Δ**	**Cohen's d**	**paired-t p**
Final Score	4.563 ± 0.233	4.508 ± 0.217	+0.055	0.262	0.097
Writing Clarity (H1a/H1b)	4.454 ± 0.137	4.360 ± 0.238	+0.094	0.342	0.030*
Utility (H2)	4.715 ± 0.148	4.621 ± 0.208	+0.094	0.417	0.009**
Accuracy	4.520 ± 0.615	4.543 ± 0.626	−0.023	−0.046	0.764

* $p < 0.05$, ** $p < 0.01$. Cohen's d is the paired-sample variant (mean of differences / SD of differences). Holm-Bonferroni-corrected p-values and Wilcoxon signed-rank robustness checks are reported in Appendix A; both the Writing Clarity test (rejecting H1a in favour of H1b) and H2 survive Holm correction under the paired-t test.

$p = 0.030$). The Flat advantage is small in absolute magnitude but, as we show in Section 4.5, its mechanism is sharper than the headline test suggests.

Table 4: Writing Clarity Per-Dimension Comparison

Dimension	**Flat (M)**	**Hier (M)**	**Δ**	**Cohen's d**	**paired-t p**
Structure	4.620	4.641	−0.021	−0.091	0.557
Coherence	4.563	4.465	+0.098	+0.192	0.220
Conciseness	4.179	3.981	+0.197	+0.441	0.007**

** $p < 0.01$. Per-dimension tests reported uncorrected; Holm-Bonferroni-corrected p-values across the six per-dimension follow-ups (WC × 3 + UT × 3) are in Appendix A. *Note:* the composite Writing Clarity gap is driven entirely by Conciseness; Structure and Coherence do not differ significantly between architectures.

Conciseness is the primary driver of the directional Writing Clarity gap ($p = 0.007$), indicating that revision loops produced wordier, more repetitive text. Structure showed no detectable difference between architectures ($d = -0.091$, $p = 0.557$).

Interpreting the Structure null. A non-significant difference test is not by itself evidence that the architectures are equivalent on Structure; it is consistent with at least three states (genuine equivalence, low statistical power for small effects, or a ceiling effect that compresses scores). To probe equivalence directly we ran a two one-sided t-test (TOST) on the paired Structure differences. At the conventional small-effect bound ($|d_z| = 0.20$, raw $|\Delta| < 0.046$), the TOST fails to reject ($p = 0.238$) and we cannot claim formal equivalence. At a more lenient bound ($|d_z| = 0.35$, raw $|\Delta| < 0.080$), the TOST does reject ($p = 0.048$), so we can claim equivalence only up to a small-to-medium-effect magnitude. A ceiling effect is also operating: 83.7% of Flat reports and 88.4% of Hierarchical reports score ≥ 4.5 on Structure, leaving little room for either architecture to differentiate. Together these observations support the weaker reading that Structure exhibits no detectable architectural difference at our sample size; they do not support the stronger reading that the architectures are positively shown to be equivalent on the

construct most directly associated with classical exploitation.

4.2 H2: Flat Coordination Produces Higher Utility

H2 is supported. The Flat architecture produced significantly higher Utility (M = 4.715 vs. 4.621, Cohen's $d = 0.417$, paired-t $p = 0.009$). This was the strongest finding and is robust to multiple-comparison correction (Appendix A).

Table 5: Utility Per-Dimension Comparison

Dimension	**Flat (M)**	**Hier (M)**	**Δ**	**Cohen's d**	**paired-t p**
Actionability	4.997	4.984	+0.013	+0.191	0.224
Root Cause Analysis	4.449	4.313	+0.136	+0.230	0.144
Strategic Depth	4.698	4.560	+0.138	+0.477	0.003**

** $p < 0.01$. Per-dimension tests reported uncorrected; Holm-Bonferroni-corrected p-values are in Appendix A.

Strategic Depth was the dimension most strongly favoring Flat ($p = 0.003$) and is the only per-dimension test that survives Holm-Bonferroni correction across the six follow-ups (Holm $p = 0.021$, see Appendix A). Because this is a single per-dimension finding from a single dataset and the rubric itself (Appendix E) was operationalised specifically for this study, we frame the Strategic Depth survival as an exploratory result whose generalisation requires cross-dataset replication. Actionability was at ceiling for both architectures (98% of scores ≥ 4.9).

Specification-extraction accuracy showed no significant difference between architectures (Cohen's $d = -0.046$, paired-t $p = 0.764$); 60–63% of reports had perfect spec extraction in both conditions. The architectural regression in Appendix C confirms that variation in specification-extraction accuracy is explained by writer-model identity and number of available reviews rather than by topology. This metric covers only the recovery of verifiable specifications; it is not a test of broader factual reliability or hallucination prevention in analytical claims.

4.3 Inter-Rater Reliability

Table 6: Inter-Rater Reliability (Krippendorff's Alpha)

Metric	**Flat α**	**Hier α**	**Interpretation**
Writing Clarity	0.084	0.167	Poor
Utility	0.218	0.299	Fair

Absolute alpha is low for Writing Clarity ($\alpha < 0.2$), driven primarily by GPT-5.4's systematic severity as a judge (mean = 3.86 vs. panel mean of 4.44–4.68); excluding GPT-

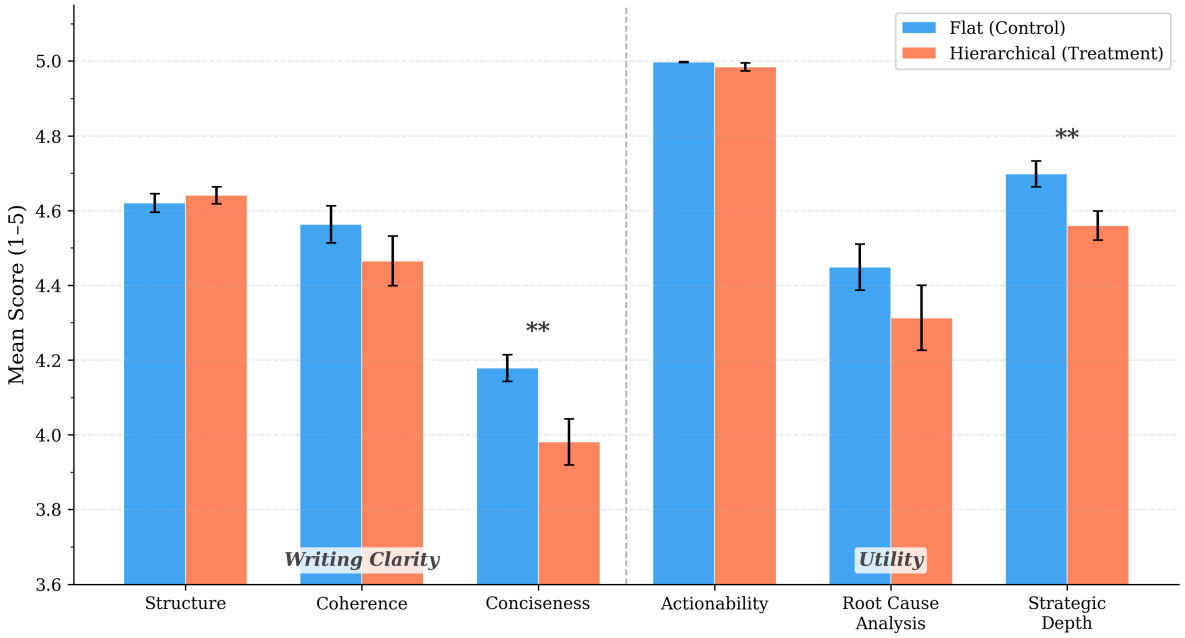


Figure 1: Per-dimension mean scores for Flat (control) and Hierarchical (treatment) on the three Writing Clarity dimensions (left) and the three Utility dimensions (right). Error bars show ±1 standard error. The composite Writing Clarity gap is driven by Conciseness ($d = 0.44$, $p = 0.007$, marked **) while Structure and Coherence show no detectable architectural difference. The Utility gap is concentrated in Strategic Depth ($p = 0.003$, marked **, Holm-surviving) with Actionability at ceiling in both arms.

5.4 raises alpha to 0.258 (Flat) and 0.341 (Hierarchical), within the range reported in NLP evaluation studies [17, 16]. Crucially, the paired design tests within-product differences rather than absolute levels: every judge agrees with the panel-mean direction on at least 68% of paired comparisons for Writing Clarity and 77% for Utility, with three of five judges exceeding 80% on both (full per-judge breakdown in Appendix B).

4.4 Efficiency Trade-offs

Table 7: Efficiency and System Overhead Comparison

Metric	Flat (M ± SD)	Hierarchical (M ± SD)	Overhead
Total Cost (USD)	\$0.504 ± 0.041	\$0.606 ± 0.189	+20.2%
Generation Cost (USD)	\$0.252 ± 0.041	\$0.354 ± 0.189	+40.5%
Latency (seconds)	209.6 ± 98.4	281.6 ± 159.3	+34.3%
Total Tokens	49,370 ± 11,254	74,781 ± 60,824	+51.5%

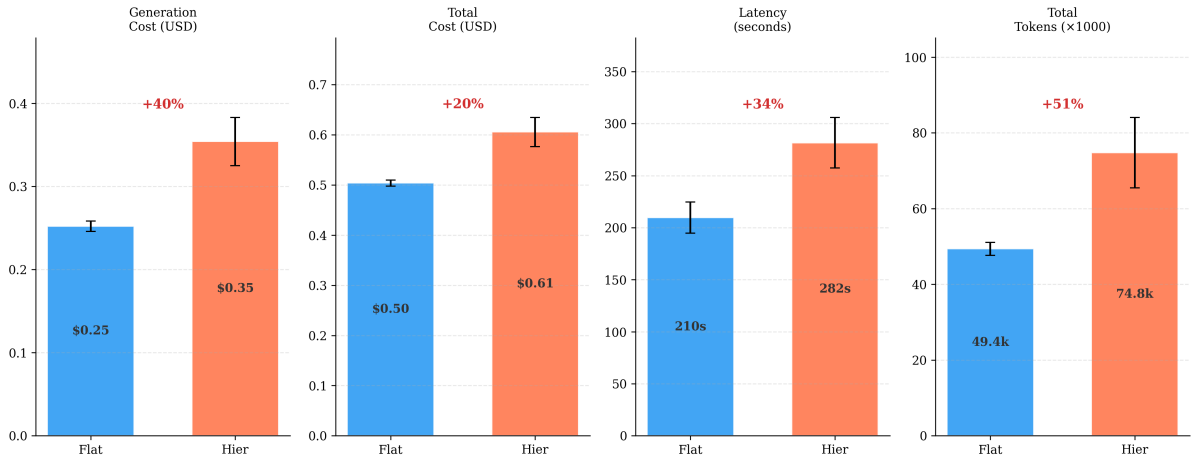


Figure 2: Efficiency comparison between Flat and Hierarchical architectures across the four cost dimensions. Hierarchical consumes 40% more in generation cost, 20% more in total cost (generation plus evaluation), runs 34% slower, and uses 51% more tokens per report, with no corresponding quality gain on any of the three quality metrics (see Tables 4, 5, and §4.2 specification-extraction accuracy).

4.5 Is the Harm from Hierarchy Itself, or from its Revision Loops?

Because Flat has loop_count hard-coded to zero by construction, the two predictors are mechanically linked and cannot be entered into a single regression together. We therefore decompose the analysis into two complementary views: a within-hierarchical dose–response and a paired comparison restricted to hierarchical runs that finished in zero loops. An architectural main-effect regression (Architecture + controls, full $N = 86$) is provided as a robustness check in Appendix C.

4.5.1 Dose–Response within the Hierarchical Arm

Table 8: Loop Count Distribution and Quality (Hierarchical Only, N = 43)

Loop Count	n	%	Mean WC	Mean Utility	Mean Final
0 loops	11	26%	4.475	4.758	4.512
1 loop	12	28%	4.391	4.571	4.442
2 loops	17	40%	4.364	4.561	4.576
3 loops (blocked)	3	7%	3.790	4.653	4.373

Note: A loop count of 3 indicates the Manager requested a third revision blocked by the MAX_LOOPS = 2 guard.

Within the hierarchical arm, the relationship between loops and Writing Clarity is monotonic and statistically robust in direction: Spearman $\rho = -0.397$ ($p = 0.008$), Pearson $r = -0.488$ ($p < 0.001$). An OLS regression of Writing Clarity on LoopCount controlling for writer model, number of reviews, and mean rating yields $\beta = -0.142$ per

loop ($p < 0.001$, $R^2 = 0.41$). For Utility, the dose–response is directionally negative but does not reach significance (Spearman $\rho = -0.265$, $p = 0.085$; OLS $\beta = -0.050$, $p = 0.18$). For specification-extraction accuracy, the loop coefficient is non-significant ($\beta = +0.129$, $p = 0.20$).

The dose-response coefficient is correlational, not causal. LoopCount is the Manager's own choice: products whose first draft was already weak attract more revision attempts, and products whose first draft was strong are released after zero loops. The coefficient therefore conflates the effect of an additional loop with the selection that produced it; the same selection caveat we apply to the zero-loop subset in §4.5.2 also applies to the per-loop coefficient itself. We address this with two robustness checks (Table 9). Dropping the 3 censored "loop = 3 blocked" observations roughly halves the within-hierarchical OLS coefficient ($\beta = -0.074$, $p = 0.038$, $R^2 = 0.21$): the direction and significance survive, but a meaningful share of the original magnitude was carried by the censored points where the Manager wanted a third revision the experiment did not allow. Recoding loop_count = 3 to 2 (capping rather than dropping) yields an intermediate estimate ($\beta = -0.132$, $p = 0.006$). The mixed-effects model on the full $N = 86$ (Appendix C) tells the same story: the architectural penalty is small and stable ($\beta_{\text{Hier}} \approx -0.09$ for both Writing Clarity and Utility), independent of how the loop-3 observations are handled. The primary causal claim therefore rests on the architectural contrast (a paired manipulation of loop-back authority); the within-hierarchical dose-response is supportive but not load-bearing on its own.

Table 9: Dose–Response Robustness: Within-Hierarchical OLS for Writing Clarity

Specification	**N**	β **(loop)**	p	R^2
Original (loops 0–3)	43	−0.142	<0.001	0.41
Recoded (loop = 3 capped to 2)	43	−0.132	0.006	0.32
Dropped (loop $\in \{0, 1, 2\}$ only)	40	−0.074	0.038	0.21

Each specification controls for writer model, number of reviews, and mean product rating. The loop = 3 cell contains only $n = 3$ observations (the third Manager-requested revision was blocked by the MAX_LOOPS guard); these are leverage-prone censored points. The DROP specification removes them; the RECODE specification reclassifies them as loop = 2; the ORIGINAL specification treats them at face value. The direction and significance survive in all three; the magnitude approximately halves under DROP, indicating that the censored points carry a sizeable share of the original slope.

4.5.2 Topology Effect Isolated at Zero Loops

Within this subset, Hierarchical is numerically *higher* than Flat on all three quality metrics, with small-to-medium effects on WC ($d = -0.302$, Hierarchical favoured) and Utility ($d = -0.421$, Hierarchical favoured); none reach significance given the reduced N, but the direction is consistent. This pattern is consistent with the dose-response evidence (§4.5.1)

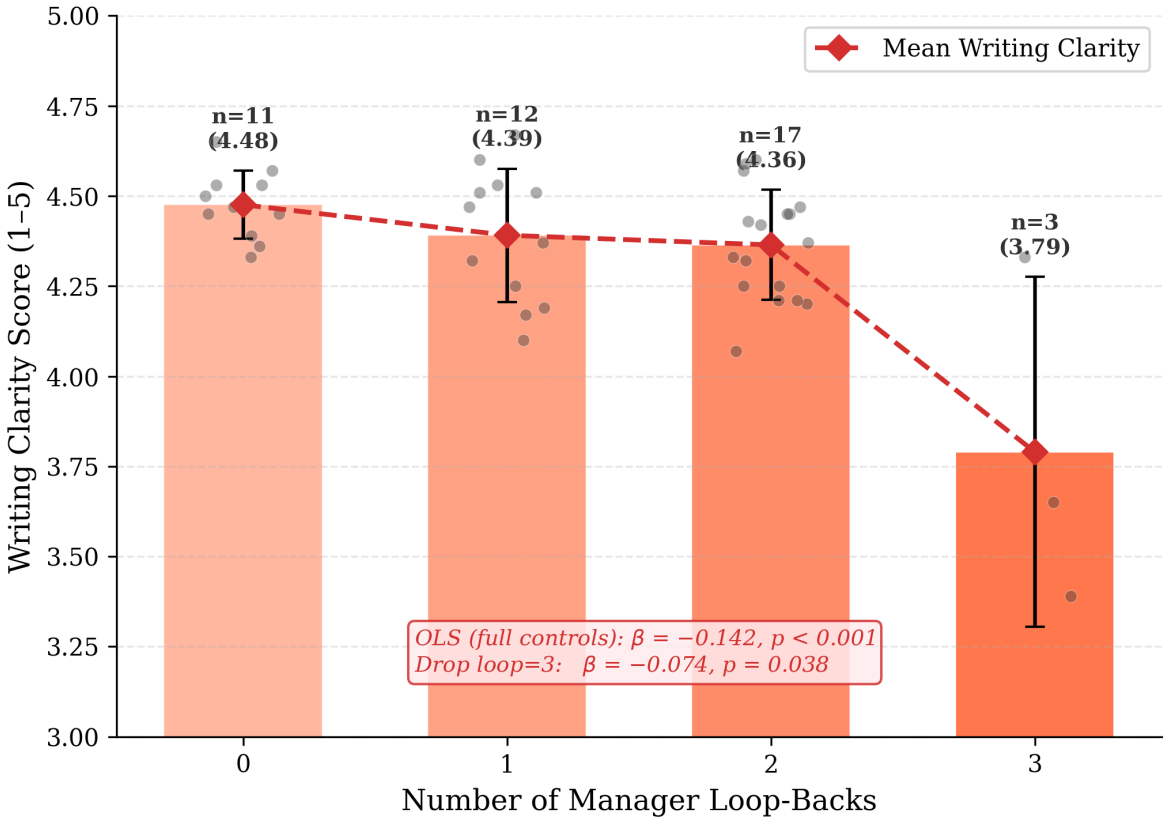


Figure 3: Within-hierarchical relationship between Manager-issued revision loops and Writing Clarity ($N = 43$ hierarchical runs). Bars show per-bucket means with ±1 SD error bars; individual products are overlaid as grey dots. Bucket sizes are $n = 11, 12, 17, 3$ for loop counts 0–3, so the loop = 3 bucket is uninformative on its own (the third Manager-requested revision was blocked by the MAX_LOOPS guard). The annotated full-controls OLS coefficient ($\beta = -0.142$, $p < 0.001$) is correlational, not causal: LoopCount is the Manager's own choice and worse first drafts attract more loops. The magnitude attenuates substantially when the three censored loop = 3 observations are dropped ($\beta = -0.074$, $p = 0.038$; Table 9). The dashed line is purely a visual guide; the loop = 3 mean should not be read as evidence of monotonic decline given the censoring and the bucket size.

Table 10: Paired Comparison, Subset Where Hierarchical loop_count = 0 (N = 11)

Metric	**Flat**	**Hier-0**	**Δ (F−H)**	**Cohen's *d***	**paired-*t* *p***
Writing Clarity	4.426	4.476	−0.049	−0.302	0.340
Utility	4.669	4.758	−0.089	−0.421	0.193
Accuracy	4.244	4.304	−0.060	−0.113	0.717

Note: Sign convention is standardised across all tables in this thesis as Δ = Flat − Hier and Cohen's d in the same direction (positive d indicates Flat scores higher; negative indicates Hierarchical scores higher). In this subset Hierarchical is numerically higher than Flat on all three metrics, so all d values are negative.

that the harm comes from the iterative-revision mechanism. The subset is not a random sample, however: these 11 products were selected by the Manager itself as requiring no revision, so we cannot rule out that they are also products where revision would have been unnecessary regardless of architecture. The within-hierarchical dose-response therefore carries the primary evidence for the mechanism, with the subset comparison serving as a suggestive triangulation rather than an independent causal test.

The two views point in the same direction. The within-hierarchical dose–response is suggestive of revision-driven harm but, as shown in Table 9, depends partly on the 3 censored MAX_LOOPS observations and is in any case correlational. The zero-loop subset is consistent with the same picture but selection-biased. The architectural main-effect mixed-effects regression on the full $N = 86$ sample (Architecture + writer-model + product-level controls, product as random intercept) is therefore the primary causal anchor: Hierarchical is associated with significantly lower Writing Clarity and Utility ($\beta_{\text{Hier}} \approx -0.09$ on both, $p < 0.05$), with the per-loop and zero-loop views supporting the same conclusion through different evidence channels. Full mixed-effects output is in Appendix C.

5 Discussion

5.1 Summary of Findings

This study tested whether Kocak et al.'s [3] predictions about influence structures transfer to LLM-based multi-agent systems. As anticipated by the competing hypotheses in §2.3, the transfer is asymmetric. The classical exploitation prediction (H1a) is rejected in favour of the LLM-specific degradation prediction (H1b): Flat scores higher on Writing Clarity than Hierarchical (Cohen's $d = 0.342$, $p = 0.030$). H2 is supported: Flat produces higher Utility ($d = 0.417$, $p = 0.009$), with the effect concentrated in Strategic Depth ($p = 0.003$). Specification-extraction accuracy showed no significant difference between architectures ($p = 0.76$). The primary causal anchor for the mechanism interpretation is the architectural mixed-effects regression on the full $N = 86$ sample ($\beta_{\text{Hier}} \approx -0.09$ for Writing Clarity and Utility, $p < 0.05$ on both; Appendix C), with two supporting views inside the hierarchical arm. The within-hierarchical dose-response (each additional loop is associated with a 0.142-point drop in Writing Clarity, $p < 0.001$) and a suggestive zero-loop subset ($N = 11$, Hierarchical numerically higher than Flat on all three metrics) both point in the direction of revision-driven harm. Both are correlational: LoopCount is the Manager's own choice (worse drafts attract more loops), and the zero-loop subset is selected by the Manager's own quality judgement. They are therefore consistent with the iterative-revision mechanism but cannot, in themselves, isolate it from the architecture manipulation. The dose-response coefficient also halves when the 3 censored

MAX_LOOPS observations are dropped (§4.5.1, Table 9). We discuss what this implies for multi-agent LLM systems research first (§5.2) and then for organizational coordination theory (§5.3).

5.2 Contributions to Multi-Agent LLM Systems Research

The current generation of multi-agent LLM frameworks (AutoGen [4], MetaGPT [6], AgentOrchestra [5]) builds on the implicit assumption that adding a managerial layer with revision authority will improve output quality, in line with how supervisory hierarchies operate in human teams. Recent surveys of LLM coordination treat hierarchical orchestration as the default architecture for complex knowledge work [32].

Our results show that this transfer does not hold for knowledge-synthesis tasks. The Hierarchical architecture consumed 51.5% more tokens, cost 20.2% more per report, and ran 34.3% slower (Table 7), yet failed to outperform the Flat baseline on any quality metric. Within the hierarchical arm, each additional Manager-triggered revision loop is associated with a 0.142-point drop in Writing Clarity ($p < 0.001$, $R^2 = 0.41$); the relationship is correlational because the Manager itself chooses how many loops to fire, and the per-loop magnitude attenuates to $\beta = -0.074$ when the 3 censored MAX_LOOPS observations are dropped (§4.5.1, Table 9). The architectural mixed-effects regression on the full sample (Appendix C) gives the primary causal anchor: Hierarchical is associated with significantly lower Writing Clarity and Utility ($\beta_{\text{Hier}} \approx -0.09$, $p < 0.05$). The loop-back authority manipulation appears benign in isolation (Hierarchical with zero loops is no worse than Flat), but the iterative revision mechanism it enables is associated with harm in this setting. Specification-extraction accuracy is governed by writer-model identity (Mistral-Large-3 $p = 0.020$) and the amount of available evidence ($p = 0.068$) rather than by topology (Appendix C). Because the metric only captures recovery of verifiable technical specifications, we do not extrapolate this null to a general claim about hallucination prevention; whether the factuality of analytical claims (which we did not measure directly) is similarly insensitive to coordination structure is an open question for follow-up work. This refines the broader question of when added coordination machinery pays for itself in LLM systems and aligns with Yang et al.'s [33] call for topology-aware multi-agent design.

For practitioners, the implication is concrete: a flat sequential pipeline should be the default, and supervisory revision loops should not be added without an explicit failure mode they are designed to fix. Returning to Simon's [2] near-decomposability principle, the value of hierarchy in LLM systems may lie in task decomposition and routing rather than in iterative quality control. A system that uses its Manager for prioritisation without granting loop-back authority might retain the structural benefits while avoiding the degradation observed here.

5.3 Contributions to Organizational Coordination Theory

Kocak, Levinthal, and Puranam's [3] framework predicts that hierarchical influence structures drive faster exploitation (cleaner convergence on a position) at the cost of getting trapped on non-peaks, while flat structures preserve exploration at the cost of slower or weaker convergence. These predictions were derived and validated computationally for human organizations.

We test the framework on LLM agent teams and find that the transfer is asymmetric. The exploration prediction (H2) holds: Flat produced significantly higher Utility ($d = 0.417$, $p = 0.009$), with the effect concentrated in Strategic Depth ($p = 0.003$, the only per-dimension test surviving Holm correction; Appendix A). This is exactly the entrapment on non-peaks that the theory warns about, although we caution that Strategic Depth is a single per-dimension result from a single dataset and the rubric–construct mapping (§3.3) is interpretive rather than externally validated. The classical exploitation prediction (H1a) does not hold at the composite level: Hierarchical does not produce cleaner writing, and the directional gap runs the opposite way, supporting the LLM-specific prediction (H1b). Within Writing Clarity, Structure (the dimension most aligned with classical exploitation in Kocak et al.'s sense of belief convergence) shows no detectable difference across architectures ($d = -0.091$). This null is consistent with the structural component of the exploitation prediction transferring, but we deliberately stop short of claiming that it does: a non-significant difference test on its own admits at least three explanations (genuine equivalence, low power for small effects, or a ceiling effect), and our equivalence-bound TOST rejects only at a lenient $\pm 0.35d$ bound and not at the conventional $\pm 0.20d$ (see §4.1). The composite Writing Clarity gap is driven by Conciseness ($p = 0.007$), meaning hierarchical reports are wordier rather than less decisive. The finding is therefore better described as a refinement than a refutation of the classical prediction: the Structure null is compatible with the structural-convergence reading without being positive evidence for it, and the iterative-revision mechanism that the theory does not explicitly model introduces a verbosity distortion that produces the directional gap at the composite level. We discuss this asymmetry in light of two conjectured, untested boundary conditions of the theory's translation to LLMs (§5.4).

More broadly, this study connects two literatures that have so far evolved separately: multi-agent LLM systems research and organizational coordination theory. It shows that predictions developed for human teams can be tested empirically on artificial teams. For organizational scholars, this opens a new empirical setting in which the mechanisms underlying classic theories can be probed with controlled architectural manipulation. For LLM-systems researchers, it provides a theoretical vocabulary (influence asymmetry, exploration vs exploitation, entrapment on non-peaks) for analysing coordination choices that have so far been treated mostly as engineering details. For cooperative-work re-

searchers, the finding extends classical organizational theory beyond human teams into the design of human-AI collaborative workflows: when an LLM agent participates as a teammate, the coordination structure must account for the fact that the same authoritative pressure that produces real deliberation in human subordinates produces sycophantic compliance in language models, so the design intuition of "add a manager" needs to be replaced with the more disciplined question of "where in the workflow is revision actually informative?"

5.4 Limitations and Boundary Conditions

The contributions above hold within the empirical regime we tested: a single short-context synthesis task (Amazon RTX 4060 gaming-laptop review synthesis), one set of five frontier LLMs in 2026, a Manager prompt drawn from a specific published framework, and a paired sample of 43 products. We do not generalise the practitioner guidelines beyond short-context synthesis of the kind tested here; whether the same pattern holds for long-context generation, code, scientific writing, or tasks where the Critic has access to richer verifiers is an empirical question we have not answered. Within this regime, the asymmetric transfer we observe is consistent with two conjectured, untested boundary conditions of the theory's translation to LLMs; we did not directly manipulate either boundary, so the conditions are inferred from the pattern of results rather than empirically demonstrated.

Embodied power relations (conjectured, untested). In human organizations, hierarchy carries genuine social and career stakes that produce real deference and self-censorship [3]. In our LLM setup, hierarchy is simulated through prompt instructions, and writer agents respond with superficial compliance: they add hedging language and padding rather than re-evaluate their reasoning, consistent with the Degeneration-of-Thought phenomenon [26] and the structural sycophancy reported under RLHF training [21, 22]. All five frontier models in our pool exhibited this pattern, suggesting it is a paradigm-level property rather than a temporary capability gap. As LLM agents acquire persistent memory, embodied tool use, and longer-horizon goals, this boundary may shift, and the exploitation predictions of Kocak et al. may become testable in ways closer to the original human setting.

Sequential vs. simultaneous communication (conjectured, untested). Kocak et al.'s flat team is modelled as a fully connected, simultaneous network. LLM agents cannot replicate this in our setup: tokens are generated sequentially, and every "interaction" is one agent reading a log of previous outputs rather than engaging in dynamic dialogue. As streaming generation matures, future work could test genuinely simultaneous LLM coordination architectures and revisit how the flat-symmetry effects predicted by the theory carry over.

Single domain and LLM-as-Judge evaluation. All experiments used RTX 4060

gaming-laptop reviews, so cross-domain replication is needed to test external validity. Quality scoring depends on a five-model LLM judge panel; although the paired design cancels judge effects that do not interact with the treatment and per-judge directional concordance with the panel mean is 68%–92% on Writing Clarity and 77%–92% on Utility (Appendix B), all judges share RLHF-style preferences and may collectively penalise an "AI style" the panel happens to dislike. Calibration against human expert judgments is left for future work.

5.5 Practical Implications

For practitioners working on short-context synthesis tasks of the kind we tested (single-document or small-corpus summarisation, review aggregation, BI-style report generation), our results yield four guidelines. We do not extend these guidelines to long-context generation, code, or tasks with richer external verifiers; their scope is the regime described in §5.4. First, default to flat pipelines for short-context synthesis: Flat matched or exceeded Hierarchical on every quality dimension while using substantially fewer resources. Second, if hierarchy is required, constrain the revision mechanism: each additional revision loop is associated with a drop of $\sim$0.142 points in Writing Clarity, and the within-hierarchical dose-response (together with the architectural mixed-effects model) is consistent with the harm coming from the iterative-revision mechanism rather than from the loop-back authority itself; consider single-pass hierarchical review or conditional loop-backs triggered only by specific failure modes (e.g., detected hallucinations). Third, invest in model selection and data quality for accuracy, not coordination complexity: specification-extraction accuracy depends on which writer model generates the report and how much evidence is available in retrieval, not on adding supervisory agents. We make no claim about the factuality of broader analytical statements, which our metric does not measure. Fourth, in human-AI hybrid teams, treat the LLM agent as a peer collaborator rather than as a subordinate to be supervised: our results suggest that the coercive review pressure that produces real deliberation in human subordinates instead produces verbose, hedged compliance in LLM teammates, which inverts the quality-gate intuition that human supervisors typically rely on.

5.6 Conclusion

The central question motivating this study was whether classical predictions about flat and hierarchical influence structures, validated for human organizations, transfer to teams of LLM agents. The short answer is that the transfer is asymmetric and the asymmetry is mechanistically interesting: exploration is preserved by flat coordination as the theory predicts, but exploitation is not delivered by hierarchical coordination in the form the theory predicts. Hierarchical agents produce wordier rather than more decisive reports, and

the harm associated with that verbosity travels through the iterative-revision mechanism rather than through the loop-back authority manipulation itself.

What is new beyond §5.1–§5.3 is the implication for how the next generation of multi-agent LLM systems should be designed and tested. The most productive follow-up direction is not to ask whether managers help, but to ask which specific revision triggers do: conditional loop-backs (revise only when a verifier flag fires), single-pass hierarchical review, or agent-to-agent negotiation without managerial authority are all tractable extensions of the present design. On the theory side, directly manipulating the two conjectured boundary conditions (embodied power and simultaneous communication) is the natural test of whether the classical exploitation prediction recovers once LLM agents acquire persistent memory and streaming generation. Until those tests are run, the contribution of this study is the disciplined claim that the human-team prediction does not translate intact to artificial teams in the regime we examined, and that the iterative-revision mechanism is the most plausible site of the divergence.

6 Data Sources and Ethics

6.1 Amazon review and product-specification data

The product specifications and customer reviews used in this study were collected via third-party data-collection services (specifically Apify and a paid commercial browser-extension scraper) during January 2026; we did not use Amazon's Product Advertising API. Collection was restricted to publicly visible product pages and customer reviews. We did not retrieve account information, IP addresses, or any non-public metadata. Review author display names that appear on the public page were not used in any analysis and are not retained in the analytic dataset. Reviews are stored only in the form needed for retrieval-augmented generation: review text, star rating, and product ASIN. We acknowledge that automated collection of Amazon page content is in tension with Amazon's terms of service for commercial purposes; the data was used solely for non-commercial academic research and is not redistributed in raw form.

6.2 AI-use disclosure

We disclose the use of AI tools in the preparation of this thesis. Claude (Anthropic) was used as a debugging assistant for the Python analysis and experimental orchestration code, and for editing-pass revision of prose during drafting. Gemini (Google) was used for literature search assistance and grammar checks. All scientific claims, statistical analyses, experimental design, hypothesis formulation, theoretical framing, and final wording are the author's own work; AI tools were used as a productivity aid for code and editing, not as a source of scientific content. The use of LLMs as agents inside the studied multi-agent

system and as the five-model evaluation judge panel is documented in §3 and §4 as part of the experimental design itself and is distinct from the editing-aid use disclosed here.

6.3 Data and code availability

Code is publicly available at https://github.com/cihatburak/Master_Thesis_Multi_Agents. The repository includes the LangGraph implementation of both architectures, the multi-model judge panel, the experiment driver, and all statistical analyses reported in this thesis. Generated artefacts (the Chroma vector store, raw model traces, and evaluation outputs) are reproducible by running the scripts in the order documented in the repository README. The Hierarchical Manager prompt is also reproduced verbatim in Appendix F; the per-dimension judge rubrics are in Appendix E. The 43-product analytic dataset is released alongside the code; raw scraped material is not redistributed for the terms-of-service reason noted above.

A Robustness Checks: Wilcoxon and Holm-Bonferroni Correction

The main text reports paired-sample t-tests as the primary tests for Writing Clarity (H1a vs H1b) and Utility (H2). As a robustness check we also computed Wilcoxon signed-rank tests and Holm-Bonferroni step-down corrections [30] for both. Tables 11 and 12 show that the substantive conclusions are unchanged: both tests survive Holm correction under the paired-t test; under Wilcoxon, the Utility test (H2) survives and the Writing Clarity test is borderline ($p = 0.051$). A specification-extraction accuracy test is included for completeness.

Table 11: Primary Tests with Wilcoxon and Holm-Bonferroni Correction

Hypothesis	**paired-t p**	**Holm-t p**	**Wilcoxon p**	**Holm-Wilcoxon p**
H2 (Utility)	0.009	0.018*	0.009	0.018*
H1 (Writing Clarity)	0.030	0.030*	0.051	0.051
Accuracy	0.764	—	0.717	—

* $p < 0.05$. Holm step-down applied across the two primary hypotheses. Specification-extraction accuracy is not corrected (reported as a supplementary observation, not a hypothesis); the label *Accuracy* in this paper always refers to specification-extraction accuracy.

Table 12: Per-Dimension Tests with Wilcoxon and Holm-Bonferroni Correction (6 dimensions)

Dimension	**paired-t p**	**Holm-t p**	**Wilcoxon p**	**Holm-Wilcoxon p**
Strategic Depth	0.003	0.021*	0.007	0.041*
Conciseness	0.007	0.033*	0.017	0.084
Root Cause Analysis	0.144	0.577	0.181	0.724
Coherence	0.220	0.659	0.489	0.977
Actionability	0.224	0.659	0.381	1.000
Structure	0.557	0.659	0.645	0.977

* $p < 0.05$. Holm step-down applied across the six per-dimension follow-up tests (WC $\times$ 3 + UT $\times$ 3).

Strategic Depth survives Holm correction under both base tests, Conciseness survives only under the t-test (Wilcoxon Holm $p = 0.084$), and the remaining four dimensions are non-significant under both. The conclusion of Section 4.2 is unchanged: Strategic Depth is the dimension on which Flat most clearly outperforms Hierarchical.

B Per-Judge Directional Concordance

Krippendorff's alpha measures absolute-level agreement and is sensitive to between-judge calibration offsets. Because our paired design tests within-product differences, we additionally report each judge's agreement with the panel-mean direction across the 43 paired comparisons.

Table 13: Per-Judge Concordance with the Panel-Mean Direction (N = 43 pairs)

Judge	**Writing Clarity**	**Utility**
GPT-5.4	68.2%	87.5%
Gemini-3.1-Pro	91.7%	91.7%
Qwen-3.5-122B	82.1%	80.0%
GLM-5	71.4%	89.7%
Mistral-Large-3	77.8%	77.1%

Every judge agrees with the panel-mean direction on at least 68% of paired comparisons for Writing Clarity and 77% for Utility, with three of five judges exceeding 80% on both. The judges therefore disagree more on absolute calibration than on the direction of architectural difference per product, which is the question the paired design is built to answer.

C Architectural Main-Effect Mixed-Effects Regression

To check that the paired-test results are not artefacts of differential writer-model assignment or product-level data availability, we fit a linear mixed-effects model on the full $N = 86$ sample with product as a random intercept and `Score` $\sim$ `Architecture + WriterModel + NumReviews + MeanRating` as fixed effects.

Table 14: Architectural Main Effect (Hier vs Flat), N = 86, 43 product groups

Dependent Variable	β **(Hier = 1)**	**SE**	p
Writing Clarity	-0.094	0.042	0.025*
Utility	-0.094	0.034	0.006**
Accuracy	$+0.023$	0.076	0.763

* $p < 0.05$, ** $p < 0.01$. WriterModel reference: Gemini-3.1-Pro. Specification-extraction accuracy is additionally predicted by WriterModel (Mistral-Large-3: $\beta = +0.650$, $p = 0.020$) and number of reviews ($\beta = +0.066$, $p = 0.068$).

Once writer-model identity, number of reviews, and mean rating are partialled out, Hierarchical is associated with significantly lower Writing Clarity ($\beta = -0.094$, $p = 0.025$) and Utility ($\beta = -0.094$, $p = 0.006$), and shows no architectural effect on Accuracy ($\beta = +0.023$, $p = 0.763$). The architectural penalty in this main-effect model is small ($\beta \approx$

−0.09) compared with the per-loop coefficient inside the hierarchical arm ($\beta = -0.142$ on WC, Section 4.5.1), consistent with the decomposition reported in the main text: the composite Writing Clarity gap is consistent with the iterative-revision mechanism doing most of the work rather than the loop-back authority manipulation alone.

D Paired-Difference Inter-Judge Correlations

Because our design tests within-product paired differences rather than absolute scores, the most informative inter-judge agreement statistic is the correlation across the 43 paired (Flat − Hierarchical) difference scores, computed separately for each judge. Krippendorff's alpha on absolute scores (Table 6) penalises calibration offsets that the paired design already cancels; the paired-difference correlation isolates whether the judges agree on the direction and magnitude of the treatment effect per product, which is the quantity our hypotheses operate on. Table 15 reports the 5×5 Pearson correlation matrices for Writing Clarity and Utility.

Table 15: Pearson Correlations of Paired (Flat − Hierarchical) Differences Across Judges (N = 43)

Panel A: Writing Clarity					
Judge	**GLM-5**	**GPT-5.4**	**Gemini-3.1-Pro**	**Mistral-Large-3**	**Qwen-3.5-122B**
GLM-5	1.00	0.36	0.44	0.13	0.42
GPT-5.4	0.36	1.00	0.32	0.10	0.26
Gemini-3.1-Pro	0.44	0.32	1.00	0.15	0.46
Mistral-Large-3	0.13	0.10	0.15	1.00	0.23
Qwen-3.5-122B	0.42	0.26	0.46	0.23	1.00
Panel B: Utility					
GLM-5	1.00	0.49	0.60	0.42	0.14
GPT-5.4	0.49	1.00	0.31	0.20	0.11
Gemini-3.1-Pro	0.60	0.31	1.00	0.37	0.15
Mistral-Large-3	0.42	0.20	0.37	1.00	0.09
Qwen-3.5-122B	0.14	0.11	0.15	0.09	1.00

Each cell is the Pearson correlation across 43 products of the paired difference (judge j's Flat score minus judge j's Hierarchical score) with the corresponding difference for judge j'. Diagonal is 1.00 by construction. Mean off-diagonal correlation: 0.29 (range 0.10–0.46) for Writing Clarity; 0.29 (range 0.09–0.60) for Utility.

The off-diagonal correlations are uniformly positive on both metrics (mean $r = 0.29$ for Writing Clarity and 0.29 for Utility), indicating that no judge runs systematically against the panel on the treatment effect. Per-judge mean differences are also directionally aligned: four of five judges score Flat > Hierarchical on Writing Clarity (range +0.026 to +0.178, with GLM-5 and Qwen-3.5 significant at $p < 0.05$), and all five do

so on Utility (range +0.047 to +0.120, with GPT-5.4 and Mistral-Large-3 significant at $p < 0.05$). The directional consistency that the paired design needs is therefore present even where absolute-level Krippendorff's alpha is low: the low alpha is driven primarily by between-judge calibration offsets, which the paired design cancels by taking within-product differences.

E Construct Validity: Inter-Dimension Correlations and Rubric Anchors

This appendix supports the interpretive caveat in §3.3 that the mapping between rubric dimensions and the exploitation/exploration constructs is an interpretive assumption of this study. We provide two pieces of evidence: a Pearson correlation matrix across the six per-dimension scores (Table 16), and the verbatim scoring rubrics issued to the LLM judges (Tables 17–18).

Table 16: Pearson Inter-Dimension Correlations (N = 86 observations, panel-mean scores)

	Struct	**Coher**	**Concise**	**Action**	**RCA**	**Strat**
Structure	1.00	0.21	0.07	0.07	−0.16	−0.04
Coherence	0.21	1.00	0.15	−0.02	0.04	−0.01
Conciseness	0.07	0.15	1.00	−0.04	0.01	0.14
Actionability	0.07	−0.02	−0.04	1.00	−0.11	0.09
Root Cause Analysis	−0.16	0.04	0.01	−0.11	1.00	0.06
Strategic Depth	−0.04	−0.01	0.14	0.09	0.06	1.00

Stacked across Flat and Hierarchical observations ($n = 86$). The three Writing Clarity dimensions are weakly inter-correlated ($r = 0.07$ to 0.21); the three Utility dimensions are weakly inter-correlated ($r = -0.11$ to 0.09); and cross-construct correlations are near zero or weakly negative. The composites are therefore averaging genuinely distinct signals rather than a tightly bundled latent construct, which strengthens the case for reporting per-dimension results alongside the composites and weakens any claim that a single per-dimension result "stands in for" the broader construct.

The construct interpretation in this study should be read with these correlations in mind. The mapping "Structure ↔ classical exploitation (belief convergence)" and "Strategic Depth ↔ exploration (divergent reasoning)" is theoretically motivated (§2 and §3.3) but not empirically validated against an external instrument; the weak inter-dimension correlations mean that a finding on one dimension does not automatically transfer to claims about the full construct, and the per-dimension results in §4 should be read as descriptive evidence in their own right rather than as proxies for the composite.

Table 17: Writing Clarity Rubric Anchors (issued to all five LLM judges)

Score	Anchor description
Structure	
5	ALL 6 required sections exist exactly as defined (Executive Summary, Specs, Feedback, Root Causes, Recommendations, Conclusion). Flawless markdown.
4	All 6 sections exist, but minor formatting issues (e.g., headers not bolded properly).
3	One section is missing entirely OR structure logic is moderately confused.
2	Two or more sections are missing. No clear separation of findings.
1	A single wall of text. No structure whatsoever.
Coherence	
5	Perfect logical flow. 0 contradictions. Findings perfectly align with the conclusion.
4	1 minor break in logical flow, but 0 factual internal contradictions.
3	1 major contradiction OR abrupt transitions.
2	Multiple contradictions. Hard to follow the narrative reasoning.
1	Completely incoherent. Random unrelated ideas strung together.
Conciseness	
5	Zero filler words. Every sentence provides new, vital data. Highly professional density.
4	Minor padding (1–2 sentences that repeat a previously stated point).
3	Noticeably verbose. Overly conversational tone unsuited for BI reporting.
2	Severe redundancy. The same exact point is made in 3 different sections.
1	Over 50% of the text is pure filler or rambling.

Each judge receives a separate prompt per dimension instructing it to (i) produce a qualitative analysis first, (ii) then assign a 1–5 score against the strict anchors above. The judge role is framed as a "Senior Editor evaluating the Writing Clarity of a Business Intelligence report."

Table 18: Utility Rubric Anchors (issued to all five LLM judges)

Score	Anchor description
Actionability	
5	ALL recommendations perfectly follow the WHO, WHAT, WHEN, IMPACT framework. Highly implementable.
4	Most recommendations follow the framework, but 1 lacks a specific owner or timeline.
3	Recommendations exist but are generally vague (e.g., "improve battery life") with no concrete steps.
2	Only 1 vague recommendation exists. Extremely difficult to implement.
1	Zero actionable recommendations provided. Pure description.
Root Cause Analysis	
5	Traces at least 2 major flaws back to manufacturing, design, or systemic origins, supported by quote evidence.
4	Traces 1 major flaw back to an underlying systemic origin.
3	Identifies what is broken perfectly, but only guesses at why without evidence.
2	Lists symptoms (e.g., "screen is bad"), but never asks or answers why.
1	Ignores negative feedback entirely. No cause analysis.
Strategic Depth	
5	Connects product flaws explicitly to business impact (brand reputation, return rates, competitor comparison).
4	Mentions business impact broadly but lacks deep strategic competitive context.
3	Purely operational focus. Reviews product features but ignores business strategy impacts.
2	Extremely shallow. Just a summary of specs without any insights.
1	Misses the point of a BI report entirely.

The judge role for Utility dimensions is framed as a "Business Strategist evaluating a BI report for practical value." The full prompt text for each dimension (including the qualitative-first protocol) is available in the project repository.

F Hierarchical Manager Prompt (Verbatim)

The Hierarchical Manager's complete prompt is reproduced below. The Flat Manager prompt is structurally identical except that loop-back authority is removed ("You do NOT have authority to reject work or force any worker to redo their task") and the workflow is fixed to a single forward pass. The full set of agent prompts is in the project repository.

```
You coordinate a BI report generation team of 4 workers:
- Researcher: Gathers product specs and customer reviews
- Analyst: Analyzes gathered data to find patterns and insights
- Writer: Compiles research into a professional BI report
- Critic: Reviews the report for accuracy and quality

All workers share a Blackboard. You must review the ENTIRE sequence of
events on the Blackboard.

WORKFLOW: The standard sequence is: Researcher → Analyst → Writer →
Critic → FINISH. After each worker completes their task, you review the
output and decide:

  1. PROCEED to the next worker in sequence (if output is sufficient), OR
  2. LOOP BACK to a previous worker (if a critical flaw is detected that
     must be rewritten)

LOOP-BACK RULES:

  • You have absolute authority to force a Loop-Back if the current output
    is corrupted.
  • You may loop back at most {max_loops} times total.
  • When looping back, provide SPECIFIC instructions on what MUST be fixed.
  • If max loops reached, you MUST proceed forward.

DECISION GUIDE:

  • After Researcher: Does the data include product specs AND customer
    reviews with quotes?
  • After Analyst: Are themes identified with evidence? Is root cause
    analysis present?
  • After Writer: Does the report follow the required 6-section structure?
  • After Critic: Has the Critic found major errors? Loop-Back. Has
    the Critic approved? Choose FINISH.
```

References

[1] Malone, T. W., & Crowston, K. (1994). The Interdisciplinary Study of Coordination. *ACM Computing Surveys*.

[2] Simon, H. A. (1962). The Architecture of Complexity. *Proceedings of the American Philosophical Society*.

[3] Kocak, Ö., Levinthal, D. A., & Puranam, P. (2022). The Dual Challenge of Search and Coordination for Organizational Adaptation. *Organization Science*.

[4] Wu, Q., et al. (2023). AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. *arXiv:2308.08155*.

[5] Zhang, S., et al. (2025). AgentOrchestra: A Hierarchical Multi-Agent Framework for Complex Task Orchestration. *arXiv preprint*.

[6] Hong, S., et al. (2023). MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. *ICLR 2024*.

[7] Vaswani, A., et al. (2017). Attention Is All You Need. *NeurIPS*.

[8] Brown, T. B., et al. (2020). Language Models are Few-Shot Learners. *NeurIPS*.

[9] Wei, J., et al. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *NeurIPS*, 35.

[10] Yao, S., et al. (2023). ReAct: Synergizing Reasoning and Acting in Language Models. *arXiv:2210.03629*.

[11] Lewis, P., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *NeurIPS*.

[12] Huang, J., et al. (2025). Language Models Do Not Have Human-Like Working Memory. *arXiv preprint*.

[13] Boussioux, L., et al. (2025). Socratic Iterative Reasoning: Enhancing LLM Decision-Making in the Beer Game Supply Chain. *ICIS 2025*.

[14] Gou, Z., et al. (2024). CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing. *ICLR 2024*.

[15] Shinn, N., et al. (2023). Reflexion: Language Agents with Verbal Reinforcement Learning. *NeurIPS 2023*.

[16] Liu, Y., et al. (2023). G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. *arXiv:2303.16634*.

[17] Zheng, L., et al. (2023). Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *arXiv:2306.05685*.

[18] Min, S., et al. (2023). FActScore: Fine-grained Atomic Evaluation of Factual Precision. *EMNLP 2023*.

[19] Liu, B. (2012). Sentiment Analysis and Opinion Mining. *Morgan & Claypool Publishers*.

[20] Doshi, A.R., et al. (2024). Generative Artificial Intelligence and Evaluating Strategic Decisions. *Strategic Management Journal*.

[21] Wei, J., et al. (2023). Simple synthetic data reduces sycophancy in large language models. *arXiv:2308.03613*.

[22] Bai, Y., et al. (2022). Constitutional AI: Harmlessness from AI feedback. *arXiv:2212.08073*.

[23] LangChain. (2024). LangGraph: Building language agents as graphs. Available at: https://github.com/langchain-ai/langgraph.

[24] Choudhury, P., Vanneste, B. S., & Zohrehvand, A. M. (2025). The Wade Test: Can Large Language Models Evaluate Strategic CEO Communication? *Working Paper*.

[25] March, J. G. (1991). Exploration and exploitation in organizational learning. *Organization Science*.

[26] Liang, T., et al. (2024). Encouraging divergent thinking in large language models through multi-agent debate. *EMNLP 2024*.

[27] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *EMNLP*.

[28] Chroma (2024). Chroma: The AI-native Open-Source Embedding Database. https://www.trychroma.com/

[29] Krippendorff, K. (2011). Computing Krippendorff's Alpha-Reliability. University of Pennsylvania.

[30] Holm, S. (1979). A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics*.

[31] Hoenig, J. M., & Heisey, D. M. (2001). The Abuse of Power: The Pervasive Fallacy of Power Calculations for Data Analysis. *The American Statistician*, 55(1), 19–24.

[32] Tran, T. H., et al. (2025). Multi-Agent Collaboration Mechanisms: A Survey of LLMs. *arXiv:2501.06322*.

[33] Yang, J., et al. (2025). Topological Structure Learning Should Be A Research Priority for LLM-Based MAS. *arXiv:2505.22467*.